

# 인터넷 분산 가상 공간에서의 효율적인 구입 및 판매 처리

고민정<sup>o</sup>, 음두헌

덕성여자대학교 컴퓨터과학부

mjko@namahe.duksung.ac.kr, dheum@duksung.ac.kr

## Efficient Processing of Buying and Selling in a Internet Distributed Virtual Place

Minjeung Ko<sup>o</sup>, Doohun Eum

School of Computer Science, Duksung Women's Univ.

### 요 약

본 논문에서는 인터넷 상에서의 물품 구입 및 판매 요구를 지원하기 위한 분산 트랜잭션 처리 모델을 제시한다. 가상 공간에서의 트랜잭션 처리를 위해서는 구입 요구와 판매 요구 조건이 부합되어야 한다. 이러한 요구 사항들을 저장하기 위해, 지리적인 지역 레벨에 따라 데이터베이스를 사용하고 이들을 DNS의 계층 구조와 비슷하게 연결한다. 또한, 각 판매 및 구입 요구 정보를 여러 사이트에 복제 저장함으로써 판매 및 구입 요구가 발생하면 부합되는 구입 및 판매 요구를 검색하는 시간을 줄일 수 있다. 판매 및 구입 요구 정보를 저장하는 방식에 따른 검색 시간에 대한 복잡도를 분석하여 제시하였다.

### 1. 서론

인터넷이 보편화되면서 웹 상에서 전자 상거래가 활발하게 이루어지고 있다. 현재는 트랜잭션 처리를 위해 대부분 중앙 집중식 데이터베이스 시스템을 사용하고 있다. 이러한 시스템은 네트워크의 특정 노드에 모든 데이터를 저장하게 되고 그 곳에서 모든 트랜잭션을 처리한다. 이러한 구조의 문제점은 다음과 같다.

1. 성능 문제 : 한 사이트에 수백만 개의 레코드를 저장하고 수 천 개의 질의를 처리하기에는 중앙 집중식 데이터베이스 시스템의 한계를 초과할 수 있다.
2. 관리 문제 : 신문사와 같은 조직에서는 이미 비슷한 목적의 데이터베이스를 구축해 놓고 있을 수 있다. 새 구성 작업을 최소로 하고 게이트웨이를 제공함으로써 이러한 데이터베이스들을 쉽게 포함할 수 있어야 한다.
3. 신뢰성 문제 : 중앙 사이트에 문제가 발생하면 전체 시스템을 사용할 수 없다.

이러한 문제들은 표준 분산 데이터베이스를 도입함으로써 어느 정도 해결할 수 있다. 분산 데이터베이스에 데이터를 배치하는 두 가지 대표적인 기술은 데이터 분할(data partitioning)과 데이터 중복(data replication)이다. 데이터 분할은 데이터가 발생한 곳에서 가까운 사이트에 저장하고 데이터 중복은 중복 데이터를 사용하는 곳에서 가까운 사이트에 저장한다[1].

인터넷의 DNS(Domain Name System)는 인터넷 도메인 이름을 IP 주소로 변환해주는 목적의 분산 데이터베이스이다. 이 시스템은 Unix의 디렉토리 시스템과 논리적으로 유사한 트리 구조로 구성되며 트리 구조의 각 노드에 연관된 데이터베이스는 서로 다른 사이트에 저장된다. 따라서 DNS는 위에서 언급한 중앙 집중식 시스템의 문제를 해결할 수 있고 구성 노드들의 자치성을 지원한다. DNS는 데이터 분할을 기본적으로 사용하지 않지만 사용 빈도가 높은 데이터(cached data)에 대해서는 데이터 중복을 사용한다[2].

본 논문에서는 판매와 구입 요구를 지원하는 컴퓨터 네트워크 상의 가상 시장을 위해 설계된 특수 목적의 분산 데이터베이스 구조를 소개한다. 이 분산 데이터베이스는 DNS 구조에 기반한다. 데이터베이스는 트리의 노드들에 분산되고 각 노드는 지리적 지역을 나타내는 도메인이 된다. 이 도메인들은 계층구조로

구성된다.

판매와 구입 요구는 항상 트리의 잎 노드(leaf node)에서 발생한다. 이러한 요구에는 트랜잭션이 처리될 수 있는 지리적 지역을 나타내는 스코프(scope)가 연계된다. 본 논문에서는 데이터베이스에 판매와 구입 요구 정보를 저장하는 두 가지 방식을 제시한다. 첫 번째 방식에서는 요구가 잎 노드에만 저장되고 두 번째 방식에서는 잎 노드에서 시작하여 스코프 내의 노드들에 이르는 경로 노드들에 저장된다. 저장과 검색을 위한 복잡도(complexity)에 대한 분석을 제시한다.

본 논문은 다음과 같이 구성된다. 2절에서는 웹 상에서의 판매와 구입에 대한 개요를 설명한다. DNS와 유사한 우리의 계층 모델에서, 판매와 구입 요구를 분산 처리하는 과정은 3절에서 설명한다. 4절에서는 분산 질의 처리 방식에 대한 경우별 분석 결과를 제시하고 5절에서는 결론은 맺는다.

### 2. 웹 상에서의 판매와 구입

가상 분산 시장은 판매자와 구입자가 인터넷 상에서 만나 트랜잭션을 수행할 수 있게 한다. 판매와 구입에 대한 요구 정보는 데이터베이스에 저장된다. 판매자는 부합되는 구입 요구를 검색함으로써 구입자를 만날 수 있고 구입자는 부합되는 판매 요구를 검색함으로써 판매자를 만날 수 있다.

우선, 판매와 구입 요구가 한 개의 데이터베이스만을 관리하는 중앙 집중식 데이터베이스 시스템으로 처리된다고 가정한다. 또한, 예로서 SELLREQUESTS(ReqNo, Location, Make, Model, Year, Price)와 BUYREQUESTS(ReqNo, Location, Make, Model, MinYear, MaxYear, MinPrice, MaxPrice)는 자동화 판매와 구입에 대한 요구 정보를 저장한 테이블들이라 가정한다.

새로운 구입 요구, buyRequest가 접수되면 다음의 SQL 문장이 부합되는 판매 요구를 검색하기 위해 사용될 수 있다. 새로운 판매 요구도 비슷하게 처리될 수 있다.

```
select * from SELLREQUESTS
where
  Make      > buyRequest.Make and
  Model     > buyRequest.Model and
  Year      > buyRequest.MinYear and
```

Year <= buyRequest.MaxYear and  
 Price >- buyRequest.MinPrice and  
 Price <- buyRequest.MaxPrice;

위의 간단한 판매 및 구입 요구에 다음의 트랜잭션 지역인 스코프(scope)를 연계한다.

판매자의 스코프 : 판매자의 스코프는 판매 요구에 의해 명시된 지역을 말한다. 이 지역은 무상 배달 지역일 수 있다.

구입자의 스코프 : 구입자의 스코프는 구입 요구에 의해 명시된 지역을 말한다. 이 지역은 무상 인수 지역일 수 있다.

또한, 판매와 구입 요구에 명시된 스코프가 사용될 때 다음의 두 가지 규칙이 적용될 수 있다.

**OR-스코프 규칙 :** OR-스코프 규칙에서, 판매 요구와 구입 요구간의 트랜잭션은 판매자의 위치가 구입자의 스코프 내이거나 구입자의 위치가 판매자의 스코프 내이면 가능하다.

**AND-스코프 규칙 :** AND-스코프 규칙에서, 판매 요구와 구입 요구간의 트랜잭션은 판매자의 위치가 구입자의 스코프 내이고 구입자의 위치가 판매자의 스코프 내일 때만 가능하다.

트랜잭션에 사용되는 위치(location)들은 지리적인 관계를 고려하여 지역(area)으로 묶일 수 있다. 위치와 지역에 관한 정보는 AREAS(AreaNo, Area, Parent) 테이블에 저장된다.

다음의 SQL 문장은 지역 USA 내의 모든 위치들을 검색하기 위해 사용될 수 있다. start with절은 시작하는 지역을 명시하고 connect by절은 Parent와 Area 사이의 관계를 명시한다.

```
select Area from AREAS
start with Area = 'USA'
connect by prior AreaNo = Parent;

구입 요구 buyRequest가 접수되면 다음의 SQL 문장이 OR-스코프 규칙 하에서 부합되는 판매 요구들을 찾기 위해 SELLREQUESTS 테이블을 검색할 수 있다. Scope 필드는 판매자와 구입자의 스코프를 나타낸다. AND-스코프 규칙인 경우도 비슷하게 처리된다.

select * from SELLREQUESTS S
where
    Make = buyRequest.Make and
    Model = buyRequest.Model and
    Year >= buyRequest.MinYear and
    Year <= buyRequest.MaxYear and
    Price >= buyRequest.MinPrice and
    Price <= buyRequest.MaxPrice and
    buyRequest.Loc IN (select Area from AREAS
                      start with Area = S.Scope
                      connect by prior AreaNo = Parent;)
```

```
OR
S.Loc IN (select Area from AREAS
         start with Area = buyRequest.Scope
         connect by prior AreaNo = Parent;)
```

### 3. 판매 및 구입 요구의 분산 처리

중앙 집중식 데이터베이스 시스템의 문제점을 해결하기 위해 분산 데이터베이스 시스템을 사용한다. 분산 데이터베이스는 컴퓨터 네트워크 상에서 여러 노드들에 물리적으로 분산되어 저장된 공유 데이터의 논리적인 집합을 말한다.

본 논문에서는 분산 데이터베이스를 그림 1과 같은 계층 구조로 표현한다. 위치와 지역에 대한 정보는 트리 계층 구조로 구성된다. 예를 들어, 세계 시장에서의 자동차 판매 및 구입인 경우, 루트 노드는 WORLD를, 그 자식 노드들은 NATIONS를, 또 그 자식 노드들은 PROVINCES를 나타낸다.

본 논문에서는, 판매 및 구입 요구 정보를 일 노드에만 저장하는 방식을 NoPath라 한다. 그림 1은 스코프 USA를 가지는 Albany로부터의 구입 요구 B1, 스코프 Eugene을 가지는

Eugene으로부터의 요구 B2, 스코프 CA를 가지는 SanJose로부터의 판매 요구 S1, 스코프 WORLD를 가지는 Victoria로부터의 요구 S2가 NoPath 방식으로 저장된 것을 나타낸다.

요구 정보가 일 노드들에 저장된 경우, 각 트랜잭션에 대해 구입자와 판매자는 OR-스코프 규칙 하에서는 전체 데이터베이스 내의 모든 일 노드들을 방문해야 하고 AND-스코프 규칙 하에서는 스코프 내의 모든 일 노드들을 방문해야 한다.

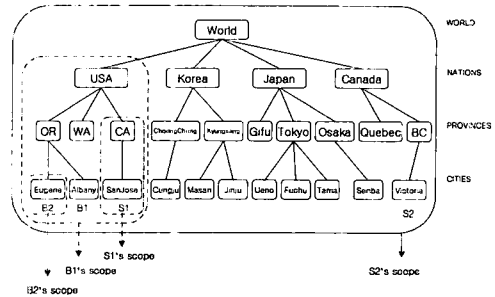


그림 1. NoPath 방식으로 저장된 요구 정보  
 반면, 요구 정보가 해당 일 노드에서 시작하여 스코프 내의 경로를 따르는 모든 경로 노드들에 저장하는 방식을 Path라 한다. 그림 2는 그림 1의 판매 및 구입 요구를 Path 방식으로 저장한 경우를 나타낸다.

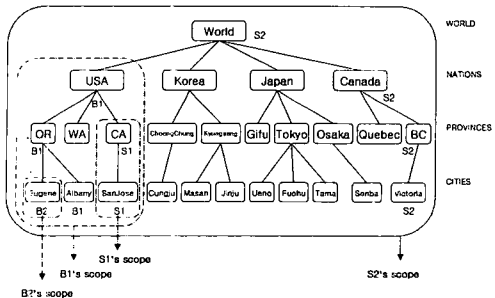


그림 2. Path 방식으로 저장된 요구 정보

Path 방식인 경우, OR-스코프 규칙 하에서는 판매자 및 구입자는 부합되는 요구를 찾기 위해 스코프 내의 모든 일 노드들을 검색하고 스코프 외는 경로 노드들만 검색하면 된다. 그림 2의 예에서, S1과 B1의 경우, S1은 무상 배달을 위해 CA 영역 내의 모든 일 노드들을 방문하고 구입자의 무상 인수를 위해 경로 노드 USA와 WORLD를 방문하면 된다. B1은 무상 인수를 위해 USA 영역 내의 모든 일 노드들을 방문하고 판매자의 무상 배달을 위해 경로 노드 WORLD를 방문하면 된다. 반면, AND-스코프 규칙 하에서는 판매자와 구입자는 스코프 내의 경로 노드들만 검색하면 된다(즉, 해당 위치에서 스코프 내의 루트까지의 경로에 있는 노드들). 예를 들어, S1과 B1인 경우, S1은 부합되는 요구를 찾기 위해 SanJose 노드와 CA 노드만 검색하면 되고 B1은 부합되는 판매 요구를 찾기 위해 Albany, OR, USA 노드들만 검색하면 된다.

따라서, 요구 정보를 경로 노드들에 저장함으로써 방문해야 하는 노드의 수를 줄일 수 있다.

### 4. 경우 분석

본 절에서는 제안한 분산 질의 처리의 여러 경우를 분석한다. 스코프규칙-구입요구저장방식-판매요구저장방식에 따라 표 1

과 같이 8가지의 경우가 있을 수 있다.

표 1. 구입 및 판매 요구에 의해 방문해야 할 노드의 수

경우	구입 요구	판매 요구
OR-NoPath-NoPath	$O(k^h)$	$O(k^h)$
OR-NoPath-Path	$O(k^{nb}+h \cdot hb)$	$O(k^h)$
OR-Path-NoPath	$O(k^h)$	$O(k^{hs}+h \cdot hs)$
OR-Path-Path	$O(k^{nb}+h \cdot hb)$	$O(k^{hs}+h \cdot hs)$
AND-NoPath-Path	$O(k^{nb})$	$O(k^{nb})$
AND-NoPath-NoPath	$O(hb)$	$O(k^{nb})$
AND-Path-NoPath	$O(k^{nb})$	$O(hs)$
AND-Path-Path	$O(hb)$	$O(hs)$

분석을 위해 h 레벨을 가지는 k-way 탐색 트리를 가정한다. 또한, 판매자와 구입자의 스코프를 각각 hb와 hs 레벨이라 한다. 각 경우에, 최악 분석(worst-case analysis)을 한다. 분석 예로 그림 3의 트리 구조를 생각한다.

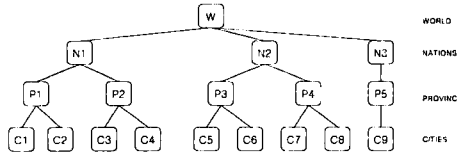


그림 3. 분산 질의 처리를 위한 트리 구조

그림 3의 트리는 WORLD, NATIONS, PROVINCES, CITIES의 4개 레벨을 가진다. 루트 노드인 W는 WORLD 레벨이고 N1, N2, N3 노드들은 NATIONS 레벨의 서로 다른 국가들을 나타낸다. 노드 P1부터 P5까지는 PROVINCES 레벨의 서로 다른 지방들을 나타내고 노드 C1부터 C9까지는 CITIES 레벨의 서로 다른 도시들을 나타낸다.

스코프 N1을 가지는 노드 C1로부터의 구입 요구 B1과 스코프 W를 가지는 노드 C8로부터의 판매 요구 S1, 스코프 P2를 가지는 노드 C3으로부터의 판매 요구 S2를 가정한다.

4.1 OR-NoPath-NoPath 경우

이 경우는 구입과 판매 요구가 발생한 잎 노드에 저장된다. 그림 4와 같이, 주어진 예에서 B1은 C1 노드, S1은 C8 노드, S2는 C3 노드에 각각 저장된다.

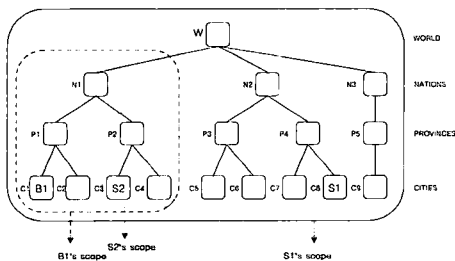


그림 4. OR-NoPath-NoPath 경우의 요구 정보 저장

구입 요구가 발생하면 잎 노드에 있는 모든 판매 요구들이 구입 조건을 만족할 수 있으므로 모든 잎 노드들이 부합되는 판매 요구를 찾기 위해 검색되어야 한다. 마찬가지로, 판매 요구가 발생하면 부합되는 구입 요구를 찾기 위해 모든 잎 노드들이 검색되어야 한다. 따라서, 두 경우 모두 방문해야 하는 노드의 수는  $O(k^h)$ 이다.

주어진 예에서, B1이 발생하면 부합되는 판매 요구를 찾기 위해 노드 C1부터 C9까지가 검색되어야 한다. 마찬가지로, S1 또는 S2가 발생하면 노드 C1부터 C9까지가 검색되어야 한다.

4.2 OR-Path-NoPath 경우

구입 요구 정보는 잎 노드에서 시작하는 경로 노드들에 저장되고 판매 요구는 발생한 잎 노드에 저장된다. 그림 5의 주어진 예에서, B1은 C1, P1, N1 노드들에 저장되고 S1은 C8 노드에, S2는 C3 노드에 저장된다.

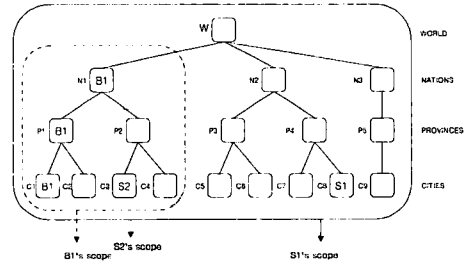


그림 5. OR-Path-NoPath 경우의 요구 정보 저장

구입 요구가 발생하면 잎 노드에 있는 어떤 판매 요구도 구입자의 조건을 만족하므로  $k^h$  잎 노드들이 검색되어야 한다. 따라서, 최악의 경우 검색되어야 할 노드의 수는  $O(k^h)$ 이다.

판매 요구가 발생하면 구입 요구는 경로 노드들에 저장되므로 구입자가 판매자의 스코프 내에 있는지 검사하기 위해 판매자의 스코프 내의 모든  $k^{hs}$  노드들이 검색되어야 한다. 또한, 판매자가 구입자의 스코프 내에 있는지 검사하기 위해 판매자의 스코프 외에 h-hs 경로 노드들이 검색되어야 한다. 따라서 검색되어야 할 노드의 수는  $O(k^{hs}+h \cdot hs)$ 이다.

주어진 예에서, B1이 발생하면 C1-C9 노드들이 부합되는 판매 요구를 위해 검색되어야 한다. S1이 발생하면 W, N1-N3, P1-P5, C1-C9 노드들이 검색되어야 하고 S2가 발생하면 P2, C3, C4, N1, W 노드들이 부합되는 구입 요구를 위해 검색되어야 한다.

5. 결론

웹 상에서의 구입 및 판매를 위한 분산 데이터베이스 모델을 제시하였다. 시스템 내의 노드들은 효율적인 트랜잭션 처리를 위해 DNS와 유사한 계층 구조로 구성된다.

본 논문에서는 스코프라는 개념을 도입하였다. 트랜잭션의 스코프란 무상 배달 및 무상 인수 또는 수입 및 수출에 가해지는 조건에 적용될 수 있는 개념이다. 제안된 모델 하에서 요구 정보를 저장하고 검색하는 기술을 설명하였다. 4절에서 보인 바와 같이 요구 정보를 잎 노드뿐만 아니라 경로 노드들에 저장함으로써 검색 시간을 줄일 수 있음을 보였다.

따라서, 우리의 모델은 (1) 노드들의 자치성을 지원하고 (2) 검색 시간을 향상시키고 (3) 향상된 신뢰성을 제공하며 (4) 분산 데이터의 용이한 관리를 통한 효율적인 트랜잭션 처리를 지원한다.

참고문헌

1. David Bell, Jane Grimson, *Distributed Database System*, Addison-Wesley, 1992.
2. Paul Albitz, Cricket Liu, *DNS and BIND*, O'Reilly & Associates Inc., 1992.
3. Daniel Florescu, Alon Y. Levy, Alberto O. Mendelzon, *Database Techniques for World-Wide Web: A Survey*, *SIGMOD Record*, vol. 27, No. 3, 59-74, 1998.
4. Serge Abiteboul, *Active Views for Electronic Commerce*, In *Proceedings of 25th International Conference on Very Large Data Bases*, 138-149, 1999.